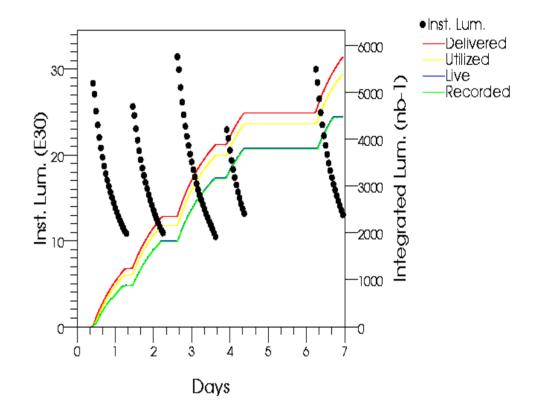
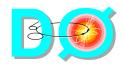


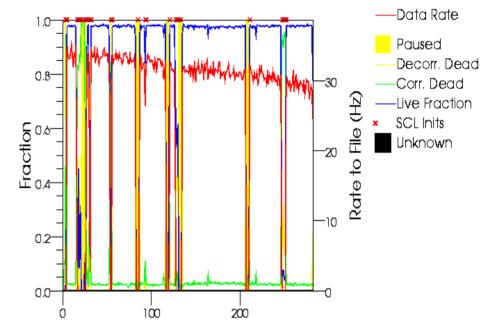
- Delivered luminosity and data taking efficiency
 - Delivered: 5.8pb⁻¹
 - Recorded: 4.8pb⁻¹ (83%)
 - ▲ Global runs 4.5pb⁻¹ (78%)
- Data taking efficiency
 - typical global runs efficiency is 85%-95%
 - quite a few special runs
 - have had runs with efficiency in the 98% range
- Number of events collected
 - 11 mln events
- Accelerator halo
 - within specifications





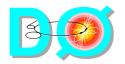
Data Taking and Triggering

- Running physics trigger list 9.50 for the last 2 weeks
 - stable
- Typical global run trigger rates
 - L1 trigger ~0.6kHz
 - L2 trigger ~250Hz
 - ◆ L3 trigger (to tape) ~50 Hz
- Currently most serious issues limiting our trigger rates
 - muon readout
 - readout code is still in process of been re-written
 - ▲ More manpower is allocated
 - Implementing detailed testing of the code before it is used for data taking
- Issues affected our downtime last week (~4 hours) total
 - No problems in excess of ~ 1 hour
 - L2 trigger downtime (problems during begin store)
- Before Sunday store lost calorimeter power supply in the hall
 - Forced us to collect mainly special runs on Sunday
 - Procedure for major problems resolution has been established



Luminosity Blocks into Run 169778

3 out of 4 best Run II runs have been collected over last week!



Summary

- D0 experiment is progressing well with physics data taking (even during Holidays...)
 - trigger list 9.50 is running on-line
 - 11 mln events collected last week
- D0 weekly data taking efficiency exceeded 80% for the first time in Run II
 - no major software/hardware problems (lack of experts did not hurt)
 - running in the "stability" region of the L1/L2 rates
 - downtime is on the level of ~5% for the week
- While finishing data collection for the winter Conferences preparations for January shutdown are progressing on schedule
 - Daily plan is developed
 - All major jobs identified and resources allocated
 - Planning to finish all major jobs by the end of 3.5 weeks shutdown

Happy New Year to everybody from DO!